**Recharge System**

**A PROJECT ON**

**“ONLINE RECHARGE SYSTEM”**

**SUBMITTED IN**

**PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF**

**DIPLOMA IN ADVANCED COMPUTING FROM CDAC**



**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY**

‘ 203, Anuda Chambers, Near Gujar Hospital,

Shaniwar Peth, Karad, Maharashtra 415110

**SUBMITTED BY:**

Lawand Uday Ramesh

**UNDER THE GUIDENCE OF**

MR. RAHUL SANSUDDI

Senior Faculty Member

Sunbeam Institute of Information Technology, KARAD

**AT**

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**

**KARAD**

**EvaluatedBySign** **CourseCoordinatorSign**

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**

**KARAD.**



|  |
| --- |
| **CERTIFICATE** |
|  |
| This is to certify that the project |

**‘Online Hostel Accommodation Portal’**

Has been submitted by

Lawand Uday Ramesh

In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing**

**(PG-DAC FEB2019)** as prescribed by The CDAC ACTS, PUNE.

|  |  |
| --- | --- |
| Place: KARAD | Date: 24-JULY-2019 |
|  | |
|  | |
|  | |
|  | |
|  | |
| **Mr. Rahul Sansuddi** |  |
| **Project Guide** | **Alumni Mentor** |

**ACKNOWLEDGEMENT**



A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Sarang Patil (Director, SIIT Karad), Mr. Prashant Lad (Course Coordinator, SIIT, Karad) and Mr. Rahul Sansuddi (Our Project Guide and Senior Faculty Member, SIIT Karad).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form. Last but not the least we thank the entire faculty, especially Mr. Nitin Jadhav, and the staff members of Sunbeam Institute of Information Technology, Karad for their support.

TEAM SUNBEAM

(DAC-15 to DAC-27)

DAC August 2005 Batch,

SIIT Karad

**INDEX**

|  |  |  |
| --- | --- | --- |
|  | **INTRODUCTION** | 1 |
|  | 1.1 Introduction | 2 |
|  | **Requirements** |  |
|  | 2.1 **Functional** **Requirements** |  |
|  | 2.2 Scope |  |
|  | 2.3 User Classes and Characteristics |  |
|  | 2.4 Design and Implementation Constraints |  |
|  | **REQUIREMENTS** |  |
|  | 3.1 Functional Requirements |  |
|  | 3.1.1 Use case for Administrator. |  |
|  | 3.1.2 Use case for Customer. |  |
|  | 3.2 Non - Functional Requirements |  |
|  | 3.2.1 Usability Requirement |  |
|  | 3.2.2 Performance Requirement |  |
|  | 3.2.3 Reliability Requirement |  |
|  | 3.2.4 Portability Requirement |  |
|  | 3.2.5 Security Techniques |  |
|  | **PROJECT DESIGN** |  |
|  | 4.1 Data Model |  |
|  | 4.1.1 Database Design |  |
|  | 4.2 Process Model |  |
|  | 4.2.1 Functional Decomposition Diagram |  |
|  | 4.2.2 Data Flow Diagram (DFD) |  |
|  | **TEST REPORT** |  |
|  | **PROJECT RELATED STATISTICS** |  |
|  | **CONCLUSION** |  |

* **INTRODUCTION TO PROJECT**

Online Recharge System is designed to offer you the opportunity to start your very own online marketplace.

Create Online recharge , it is a Recharge system designed to cater to your businesses needs and offer you the most customizable platform for multi Type Recharge System.

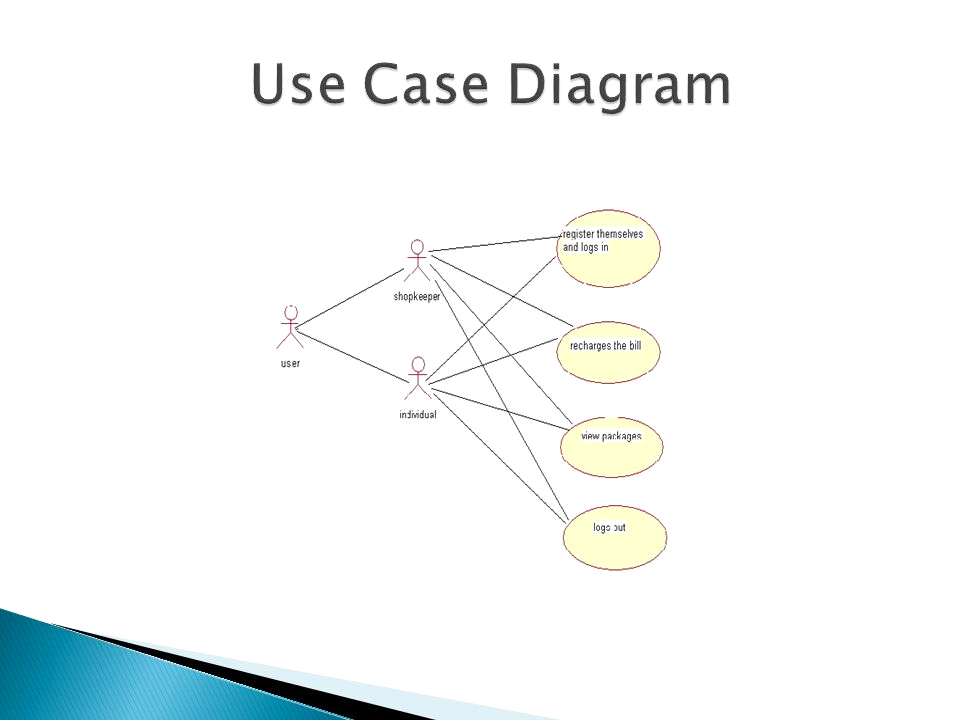
With using the multi-Recharge application, you will able to enjoy benefits including:

Flexibility to create multiple recharges, Direct link in accessing each Company recharge, Features that allow you to make profit in short and long run, Ability to operate multiple stores and spread your business, Achieve huge prospects and make more deals

**2.REQUIREMENTS**

**2.1FUNCTIONALREQUIREMENTS**

**2.1.**



**U1:Register User and Admin**

In this project registration is provided for user as well as admin

Online Recharge System. User and admin has different functionalities, prior to Registration they can view their respective list.

**Scenario1:Mainline Sequence**

**1.User:View the homepage of Online Recharge**

**2.System:Redirectuser to registration page and login page3Admin:Manage the details of user**

**U2:Select List and provide the appropriate options**

After the registration of user and admin they have be provided various options.

**Scenario1: Mainline Sequence**

**1 .System: Display the particular Recharge details after select in git.**

**2.1.2Admin registration and further details**

After admin has Login redirection is given to login page and various option are display to admin such as Display Vendor Information & Student information.

**2.1.3User registration and further details**

After user has registered redirection is given to login page and various option are display to user such as Hostel list, Selection and booking of hostel.

**2.2.5 Control**

The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

**2.2.6 Security**

The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

|  |  |  |  |
| --- | --- | --- | --- |
| **2.2.6OtherRequirements** |  |  |  |
|  | **HARDWARE** | **:** |  |
|  | **Processor** | **:** | **Pentium2.4GHzorabove** |
|  | **Memory** | **:** | **256MBRAMorabove** |
|  | **Cache Memory** | **:** | **128KBorabove** |
|  | **Hard Disk** | **:** | **10GBorabove[atleast3MBfreespacerequired]** |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Operating System** | **:** | **Windows 10** |
|  | **Font-End Tool** | **:** | **Angular7usingbootstrap** |
|  | **Back-End** | **:** | **Ms-SQL,ASP.NET MVC** |

**ASP.NET MVC**

The basic need of introducing MVC was to make complex application development easy. MVC is a lightweight highly testable framework as compared to traditional ASP.NET Web Forms. MVC focusses on Separation of Concerns. The purpose of MVC is to separate the content from the presentation and data processing from content. One thing that MVC has done is to separate the view from the code i.e., unlike Web Forms where “\*.aspx” is attached to “\*.aspx.cs” here in MVC View is a separate entity entirely.

**IIS**

Internet Information Services (IIS) 7 and later provide a request-processing architecture which includes:

The Windows Process Activation Service (WAS), which enables sites to use protocols other than HTTP and HTTPS.

A Web server engine that can be customized by adding or removing modules.

Integrated request-processing pipelines from IIS and ASP.NET.

**ANGULAR**

Angular is a type script based open source , front end web application framework led by the Angular Team at Google and by a community of individuals and corporations.

**Visual Studio 2015**

Visual Basic 2015 is the version of Visual Basic introduced by Microsoft in 2015. Microsoft has added many new features particularly those features for building mobile applications. Visual Basic 2015 is a full-fledged Object-Oriented Programming (OOP) Language implemented in the .NET Framework.  Visual Basic 2015 is bundled together with other Microsoft Programming languages C++, F#, JavaScript, Python and more in an IDE called Visual Studio Community 2015.

**Microsoft SQL Server**

SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database.

SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

T-SQL is a Microsoft propriety Language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

SQL Server Management Studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.

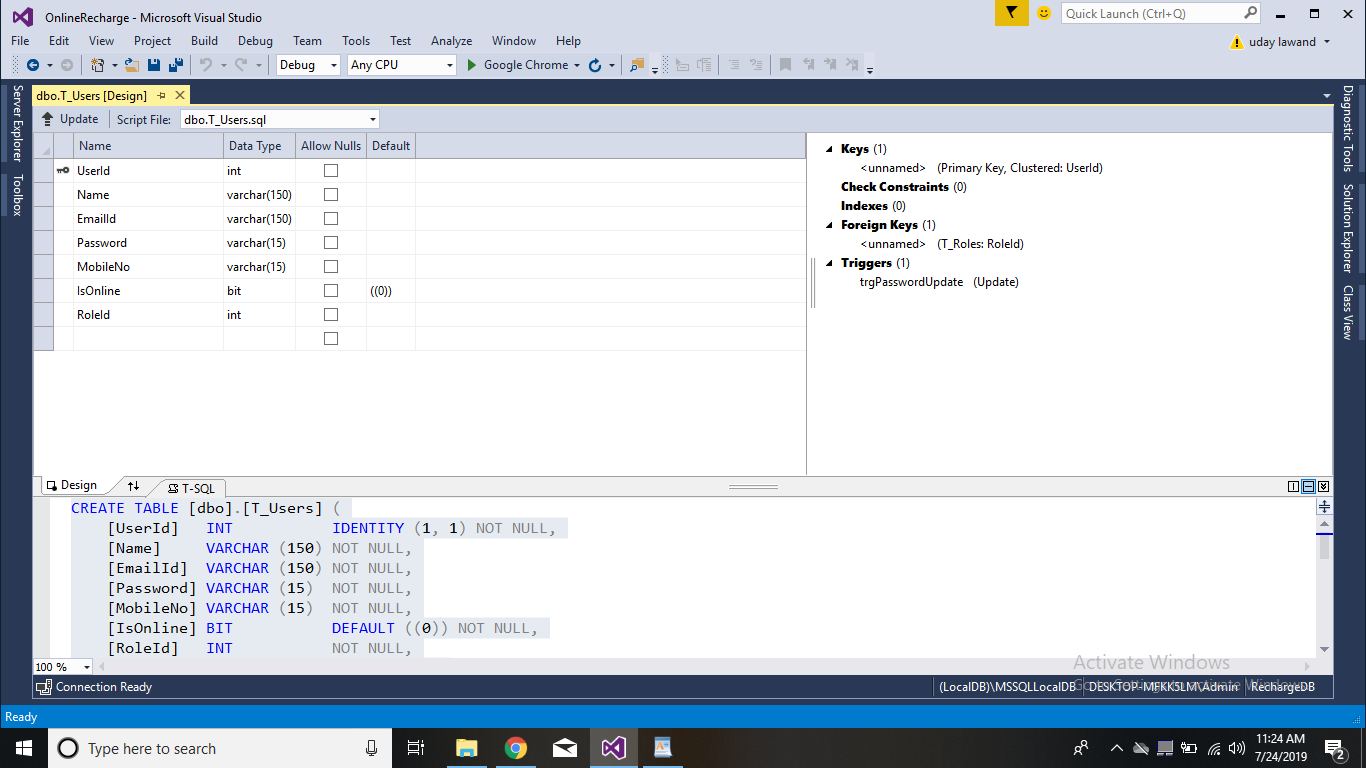
**3.DESIGN**

**3.1 Database Design**

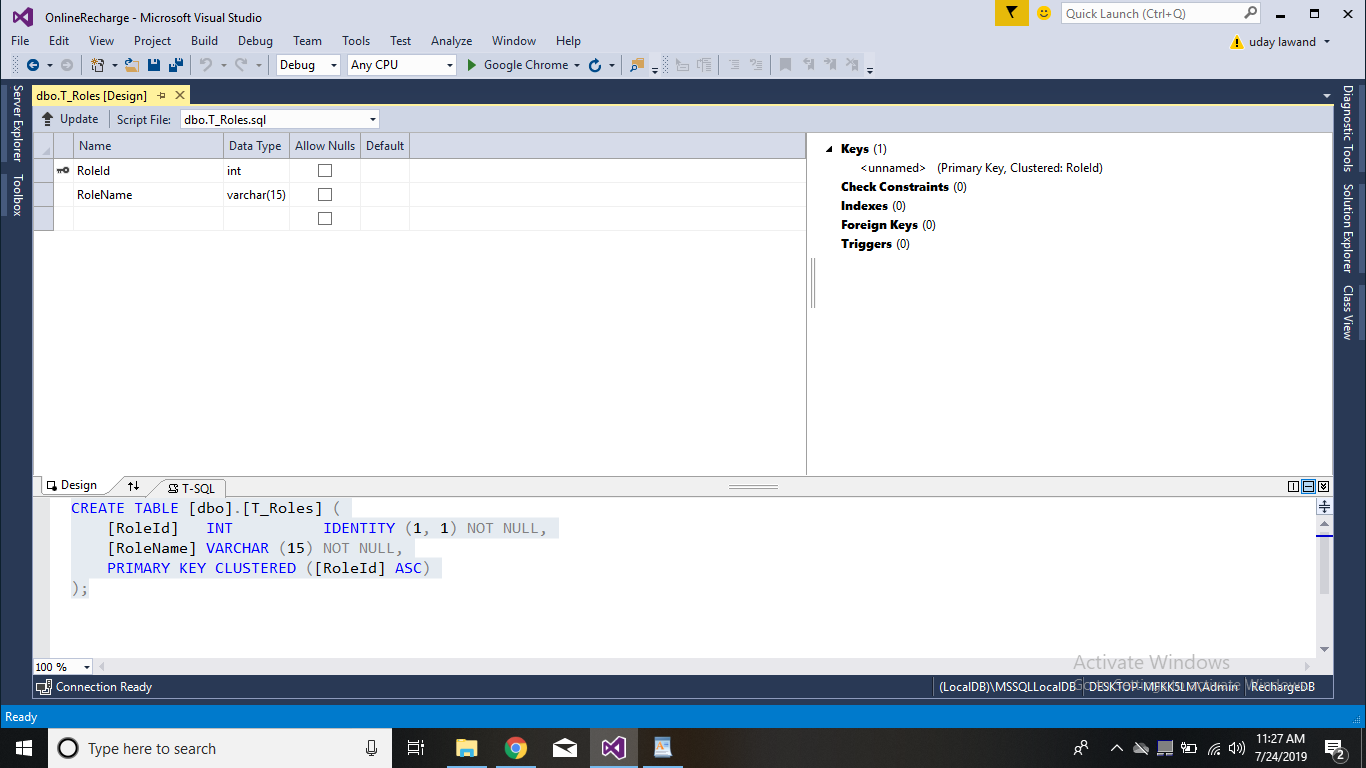
**The following table structures depict the database design.**

**Tables**

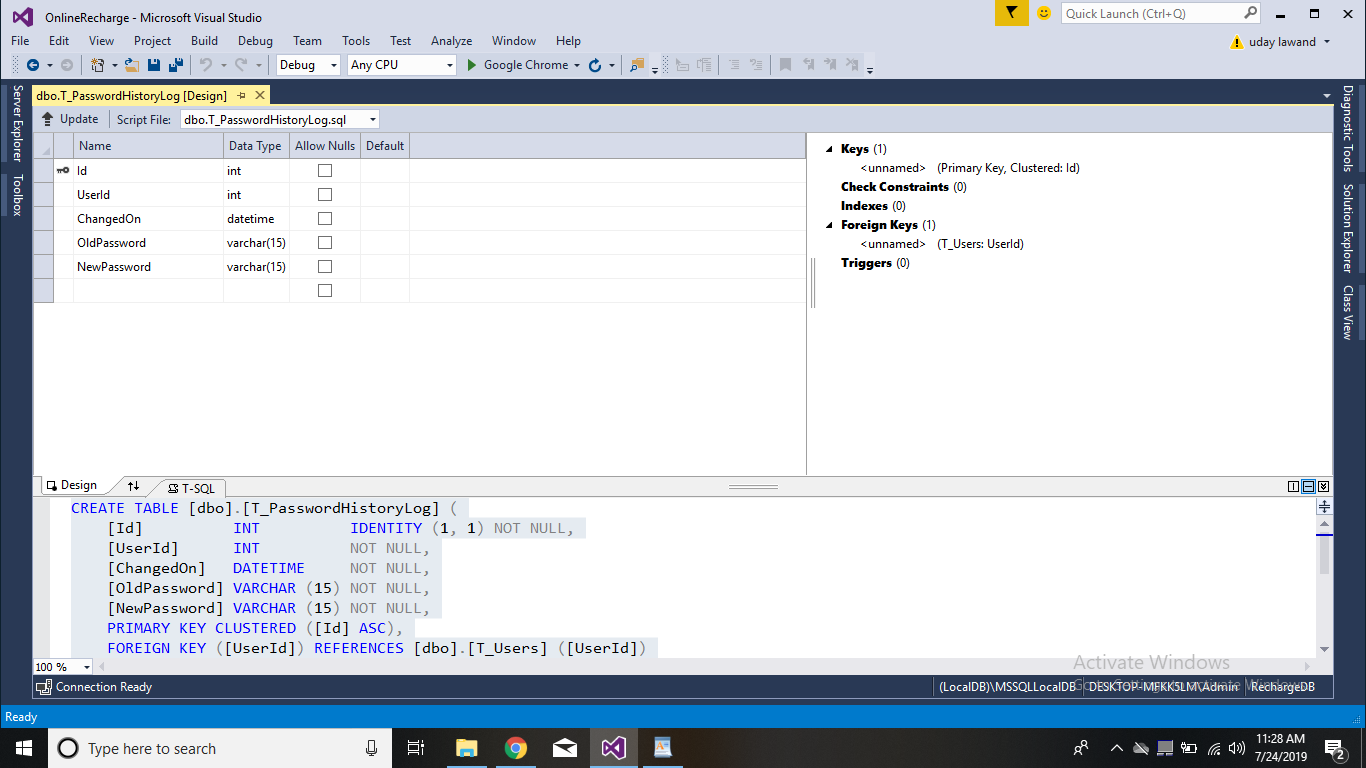
1.Customer Registration



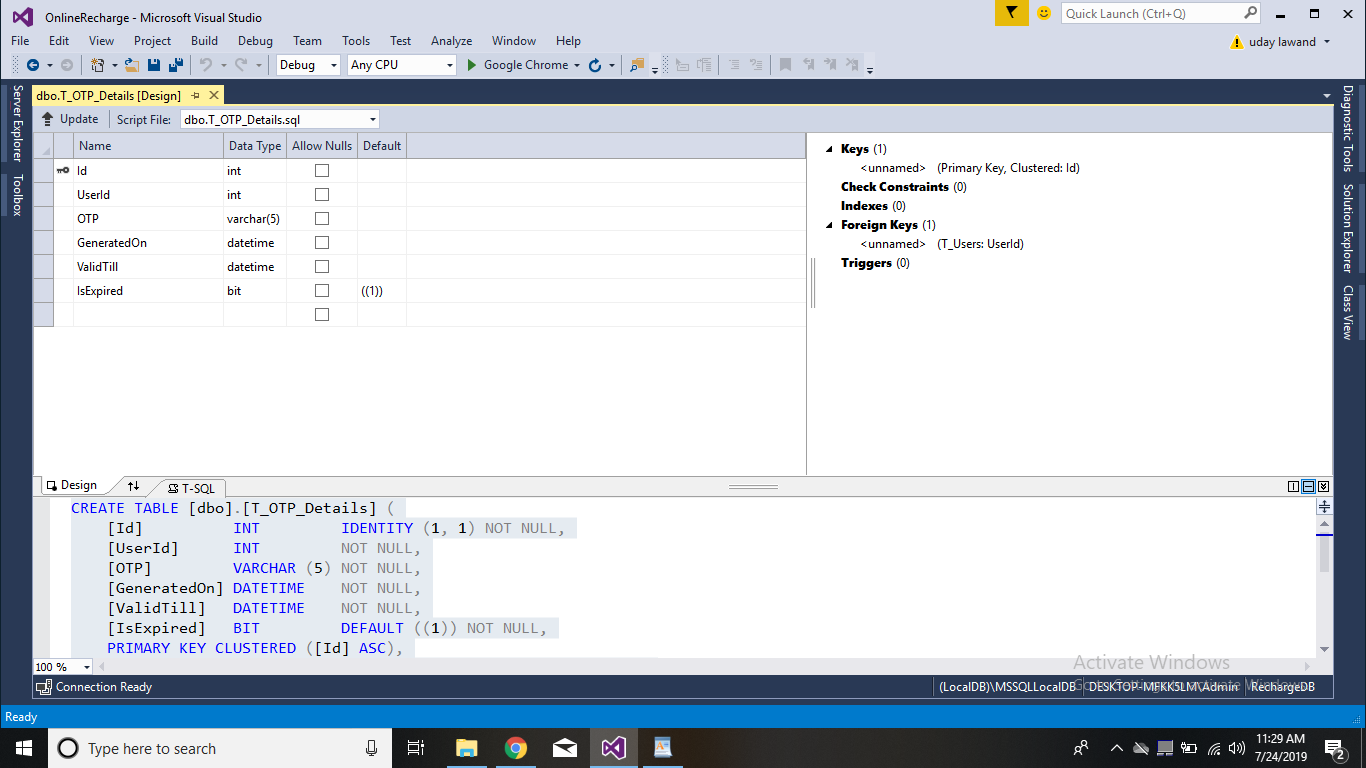
2 Table Role



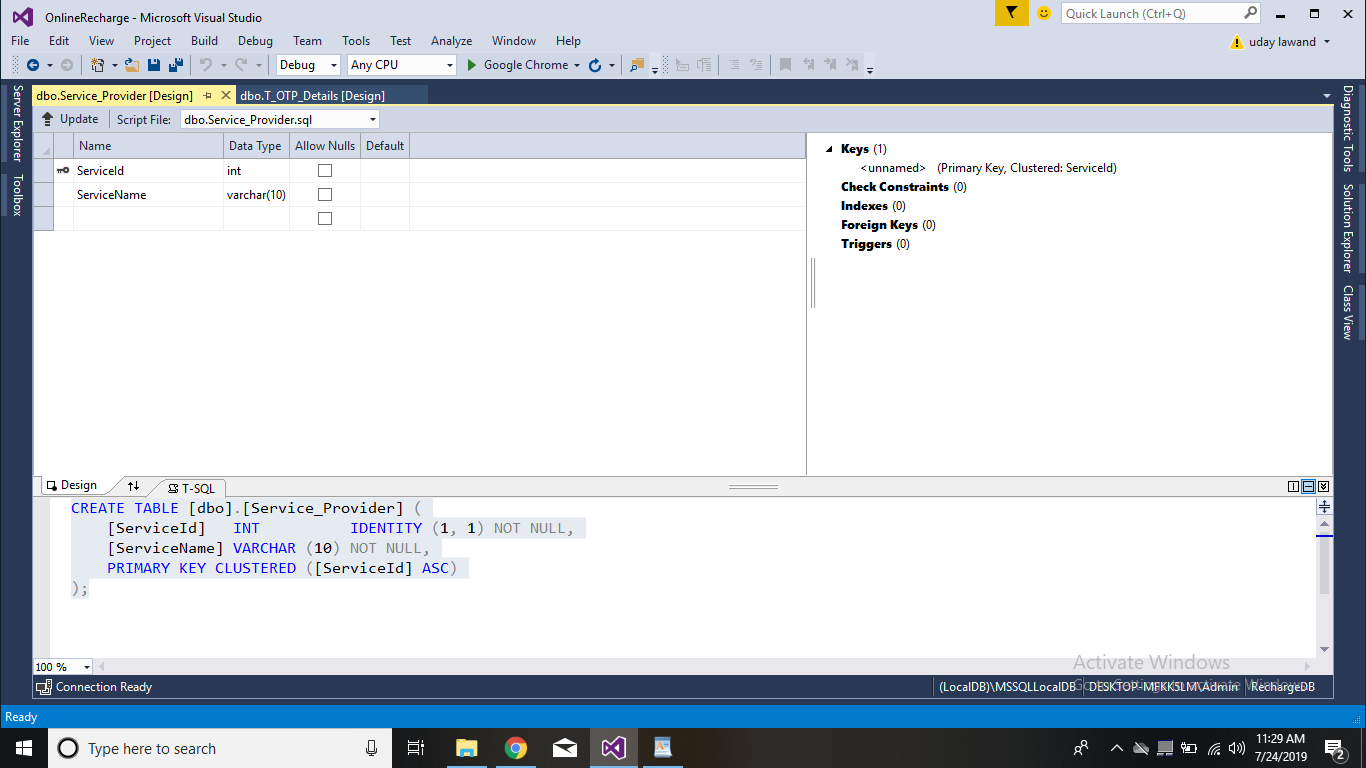
3 Table Password History



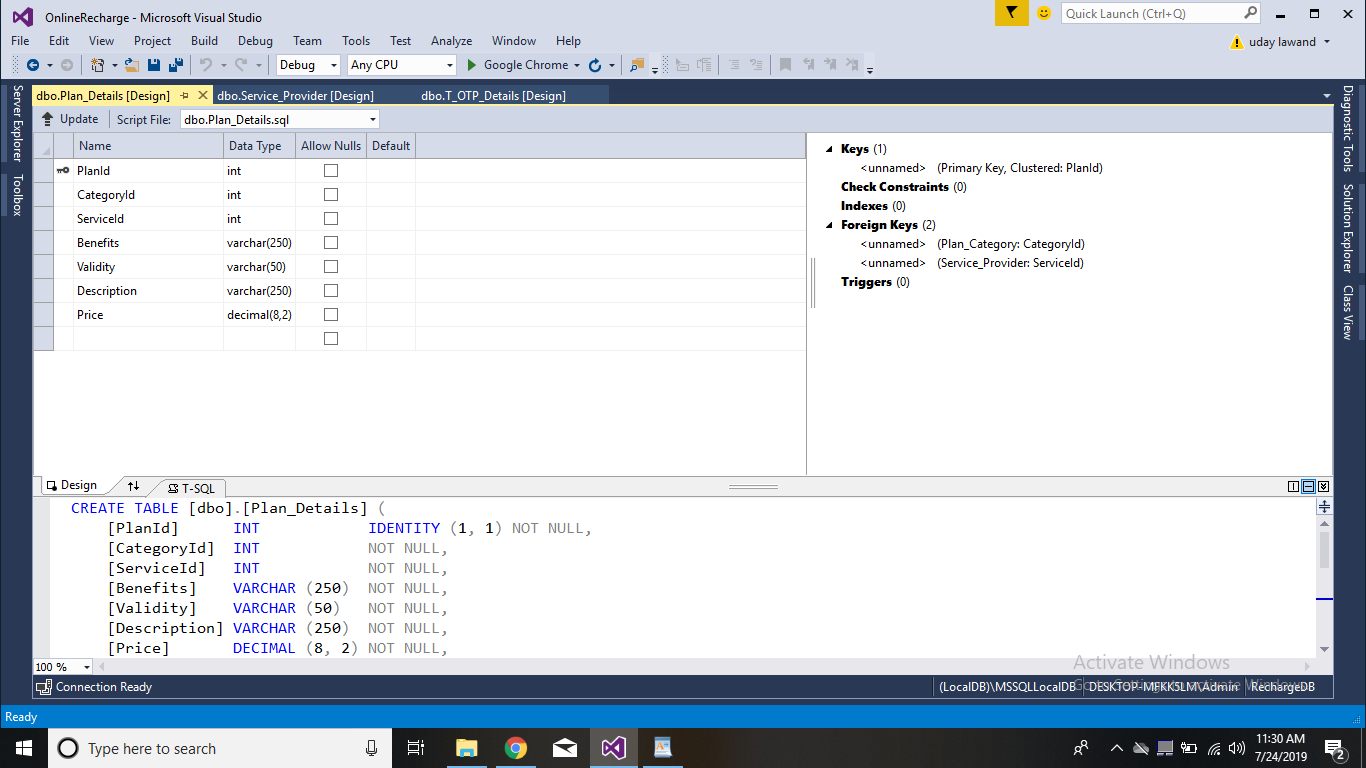
4 Table OTP Details



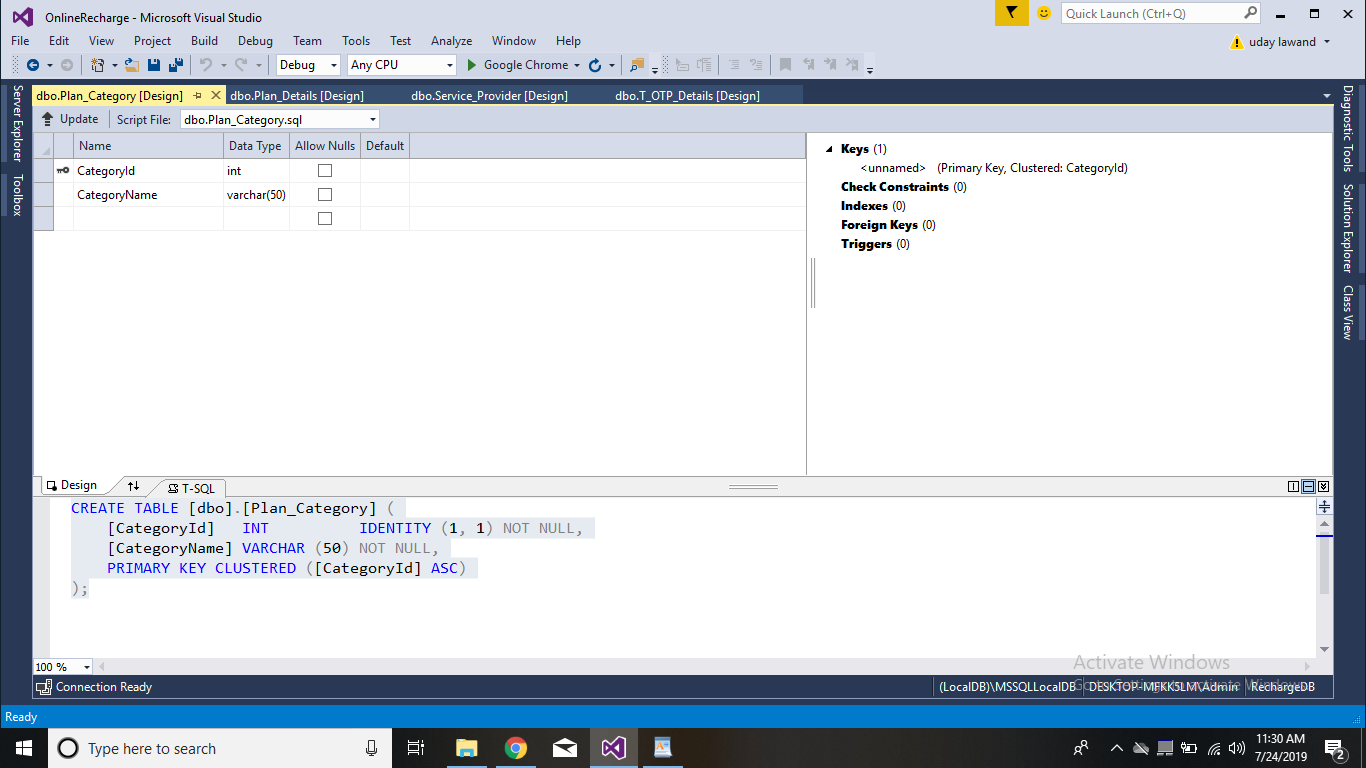
5 Table Service Provider



6 Table Plan Details



7 Table Plan Category



8 Table History



**3.2 System Design:**

What is a Methodology~~?~~

Software engineering is carry out of using preferred procedure techniques to progress the quality of a software development effort. A methodology is defined as a collection of procedures, techniques, tools, and documentation aids which will help developers in their efforts (both product and process related activities) to implement a new system. For successful implementation, a well-organized and systematic approach is crucial. Therefore, several methodologies were developed to encourage the systematic approach to planning, analysis, design, testing and implementation. Methodologies offer various tools and techniques to assist in analysis, design and testing in terms of detailed design of software, data flowcharts and database design.

**Why Methodology?**

1. To complete a project within time and budget with the expected scope and quality we need methodologies which provide for a framework.
2. Most methodologies have a general planning, developing and managing stages in common. They suggest the development team the ways of thinking, learning and arriving at a regular feasible solution.
3. To select an ideal methodology was based on project requirements and goals.

**4.** Functional Decomposition**:** The methodology should have stages according to theinterrelated activities which can be grouped into different functional areas.

**5**. Requirement Changes**:** If required, methodology provides scope to change therequirement.

1. Manage Risks**:** Determined the risk is an important activity to develop a project.

**Documentation: Methodology provides support for large documentation.**

**Analysis and Design Support:** A well-defined structure of the methodology helps foranalysis and designing to development process.

**Implementation:** The system should be implemented as per plan.

**Testing Support:** More testing, more reliable the product is.

**Object Oriented Approach:** Object oriented concepts will be used in developing theproject as it supports component reusability.

**Suitable Methodologies:**

**Waterfall Methodology:**

All projects can be managed better when segmented into a hierarchy of chunks such as phases, stages, activities, tasks and steps. It follows a linear structure starting from requirement analysis, through design, implementation and maintenance. Most widely accepted methodology for student projects, this model has been well tried and tested. Each phase of it has sub phases which produce deliverables. Requirements are fixed at initial stages before proceeding with development plans in system development projects; the simplest rendition of this is called the "waterfall" methodology, as shown in the following figure:

The graphic illustrates a few critical principles of a good methodology:

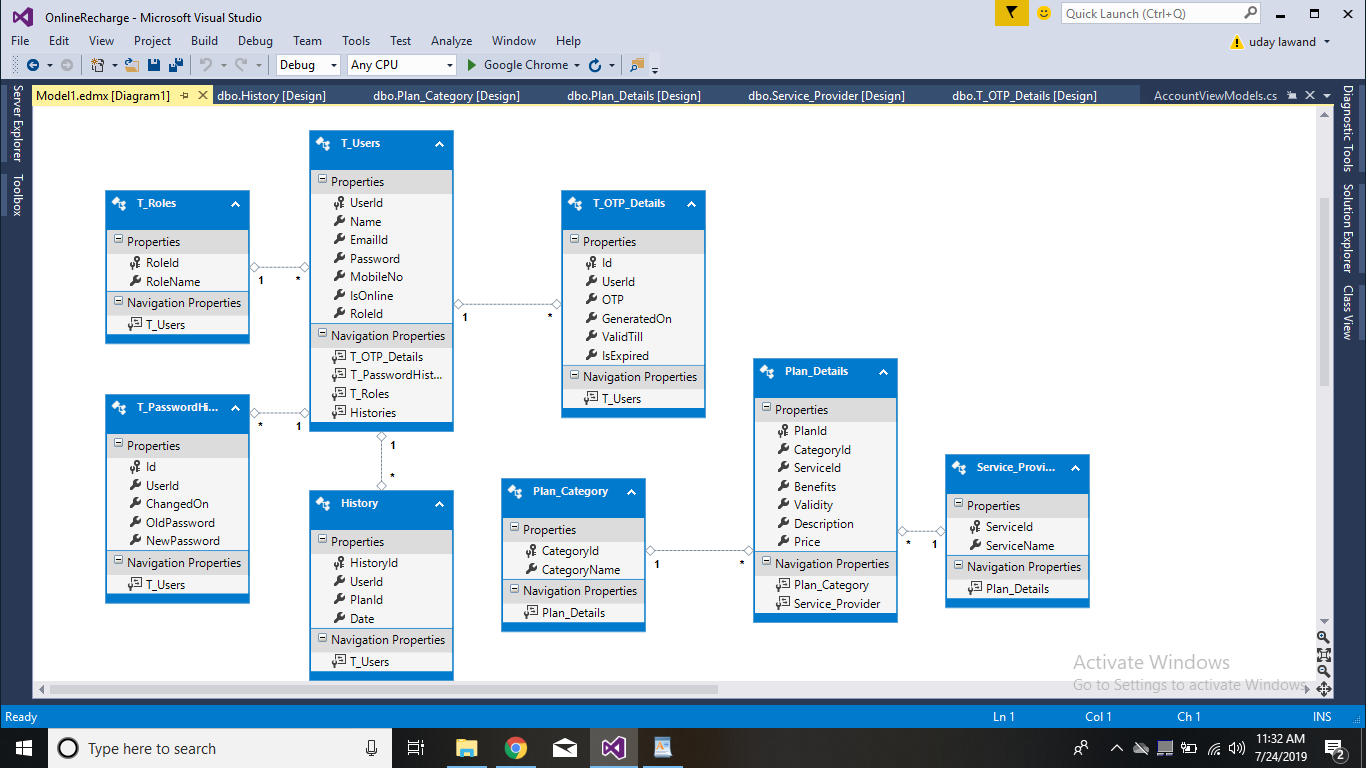
Work is done in stages,

Content reviews are conducted between stages, and

Reviews represent quality gates and decision points for continuing.

The waterfall provides an orderly sequence of development steps and helps ensure the adequacy of documentation and design reviews to ensure the quality, reliability, and maintainability of the developed software. While almost everyone these days disparages the "waterfall methodology" as being needlessly slow and cumbersome, it does illustrate

**3.3 E-R Diagram and Class Diagram:**



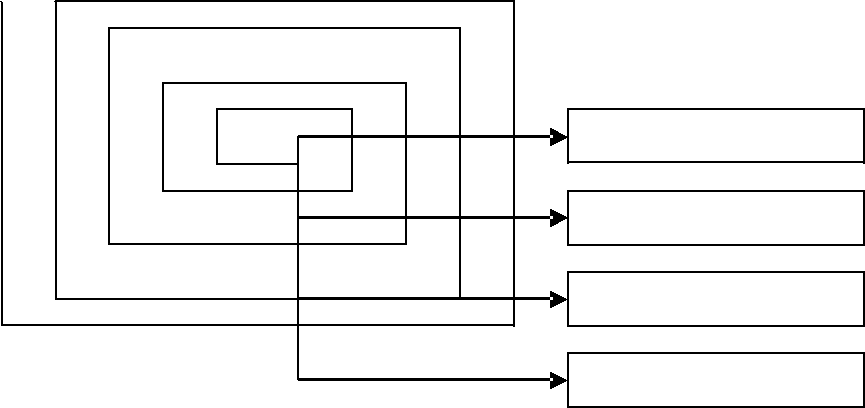
**4. TESTING PHASE**



One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.

No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

**LEVELS OF TESTING:**



**UNITTESTING**

**VALIDATION**

**INTEGRATION**

**SYSTEM**

**4.1. TEST REPORT**



**The report of the testing is given here under.**

**GENERAL TESTING**

|  |  |  |
| --- | --- | --- |
| **SR-NO**  **1**  **2** | **TEST CASE**  **User** **Login** **Page**  **Admin** **Login** **Page** | **EXPECTED RESULT**  **Redirected** **to** **Next** **page** **and** **show** **list**.  **Redirected** **to** **the** **next** **page** **showing** **various**  **options**  **Displays** **Hostel List and add hostel link.**    **Displays options such as Food type,view and**  **Details,view** **vendor and student**  **details**  **Admin** **can approve or reject vendor.** |
| **3**  **4** | **Vendor Login Page** |
| **5** | **Admin** **page** |
| **6** | **Approve/Reject** |
| **7** | **Sign** **in** **Page**  **(Registration)**  **View** **HomePage**  **Log** **out** | **All** **Fields** **should** **be** **display** **dynamically,** **filled** **for**  **submission** **and** **redirected** **to** **login** **page.**  **Without** **login** **user** **can** **only** **view** **organisations**  **After** **log** **out** **index** **page** **should** **be** **display** |
| **8**  **9** |

**6. PROJECT MANAGEMENT RELATED STATISTICS**

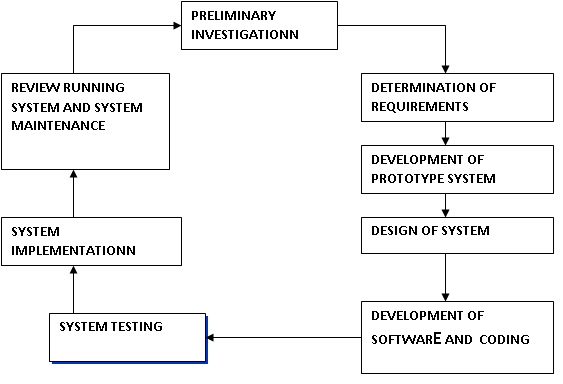


|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **WORK** **PERFORMED** | **SDLC** **Phase** | **Additional** **Notes** |
| **MAR** **12 19** | **Project** **Allotment** **and** **User** **Requirements** **Gathering** | **Feasibility** **Study** | **Performed** **Literature** **Survey** **to** **collect** **requirements.** |
| **JULY** **14,2019** | **Initial** **SRS** **Document** **Validation** **And** **Team** **Structure** **Decided** | **Requirement** **Analysis** **(Elicitation)** | **The** **initial** **SRS** **was** **finalize** **to** **understand** **requirements** **better** |
| **JULY** **15,2019** | **Designing** **the** **use-cases,** **Class** **Diagram,** **Collaboration** **Diagram,** **E-R** **Diagram** **and** **User** **Interfaces** | **Requirement** **Analysis** **&** **Design** **Phase** | **Database** **Design** **completed** |
| **JULY** **16,2019** | **Collaboration** **Diagram,** **E-R** **Diagram** **and** **User** **Interfaces** | **Design** **Phase** | **---** |
| **JULY 16,2019** | **Business** **Logic** **Component** **design** **Started** | **Design** **Phase** | **---** |
| **JULY 16,2019** | **Coding** **Phase** **Started** | **Coding** **Phase** | **---** |
| **JULY 21,2019** | **OFF** | **OFF** | **OFF** |
| **JULY 17,** **2019** | **Implementation** **of** **Web** **Pages.** | **Coding** **Phase** | **40%** **of** **Class** **Library** **implemented.** |
| **JULY 18,** **2019** | **Implementation** **of** **Web** **pages** **and** **Business** **Logic** | **Coding** **Phase** **and** **Unit** **Testing** | **Class** **Library** **Development** **going** **on.** |
| **JULY 18,** **2019** | **Implementation** **of** **Business** **Logic** | **Coding** **Phase** | **Class** **Library** **Modified** **as** **per** **the** **need.** |

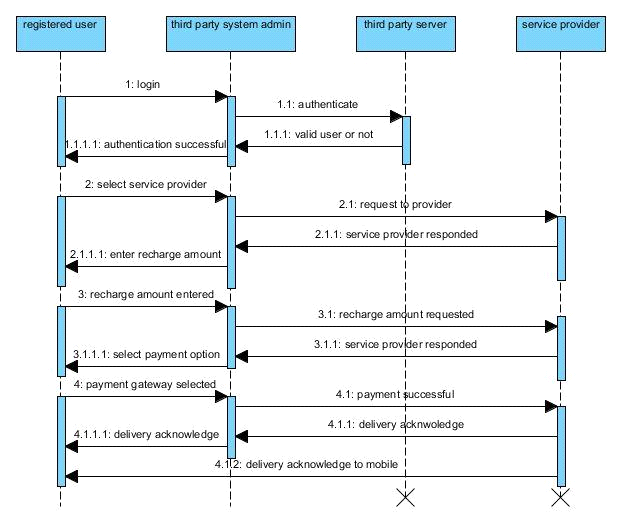
**Appendix A**



The different phase of **Software Development Life Cycle** is shown below.

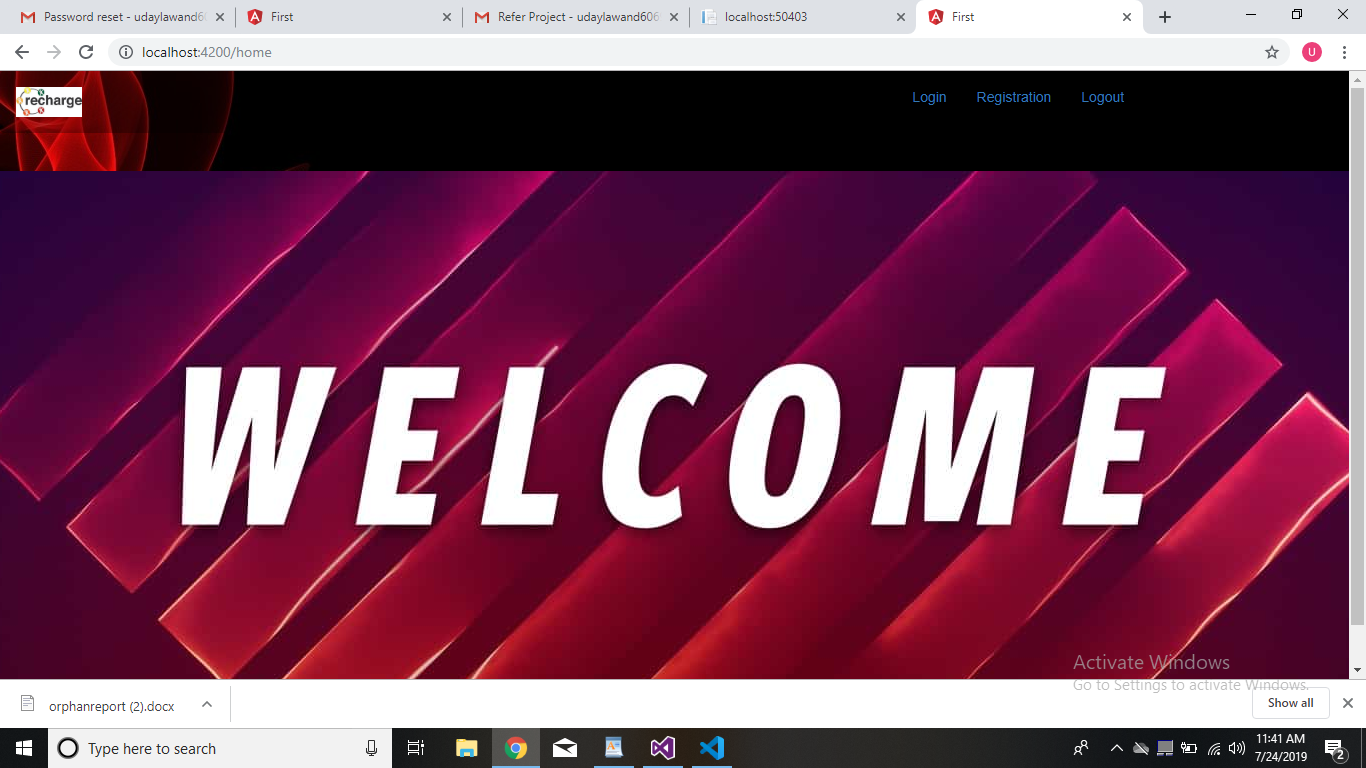


**User Sequence Diagram**

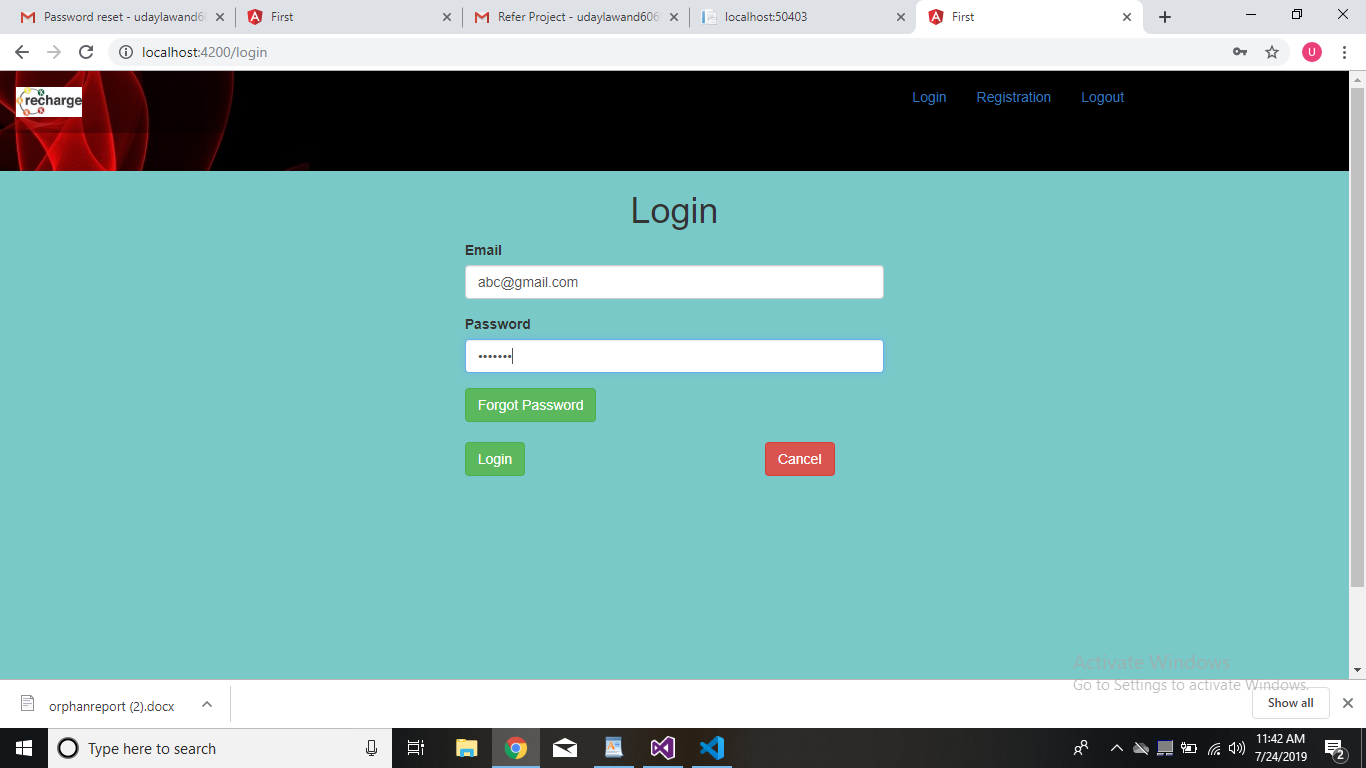


**Appendix B**

**Interface1 : Login Page:**



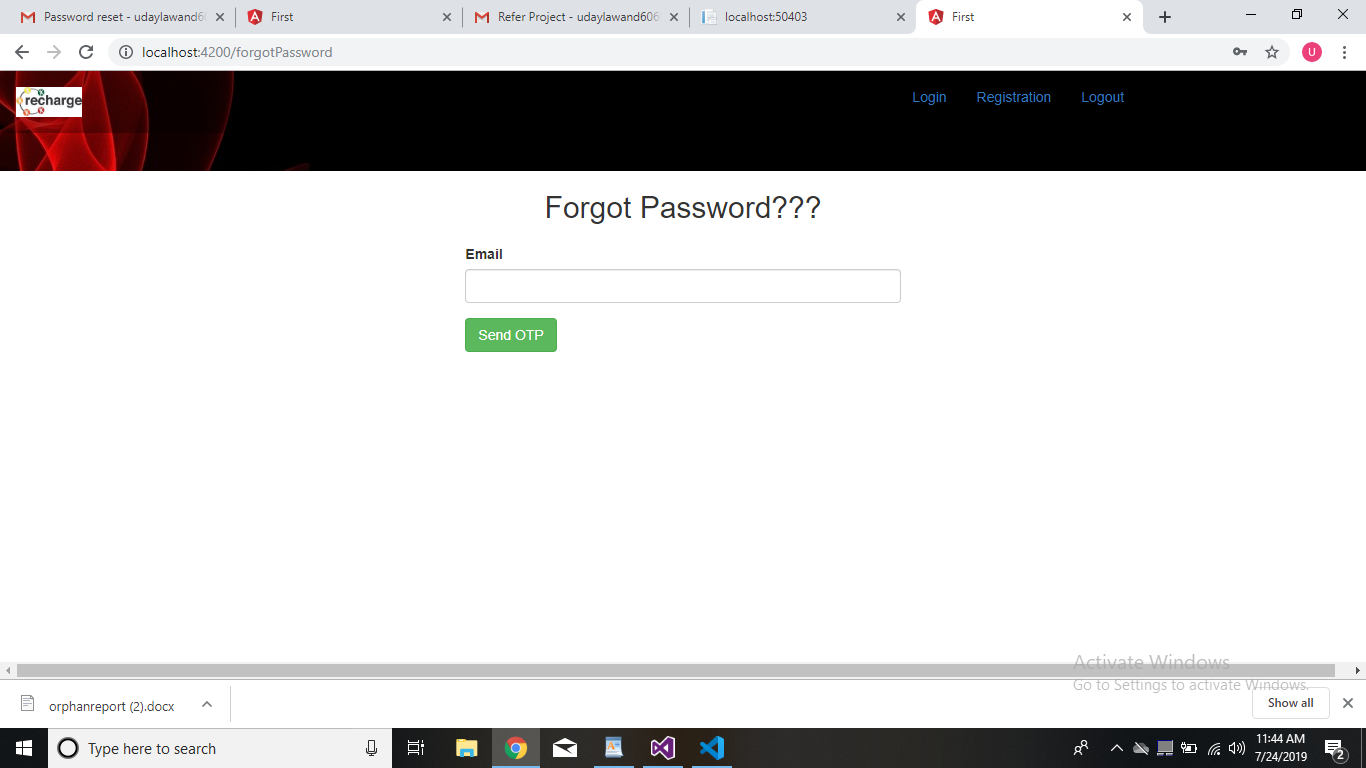
**Interface2:Login Page:**



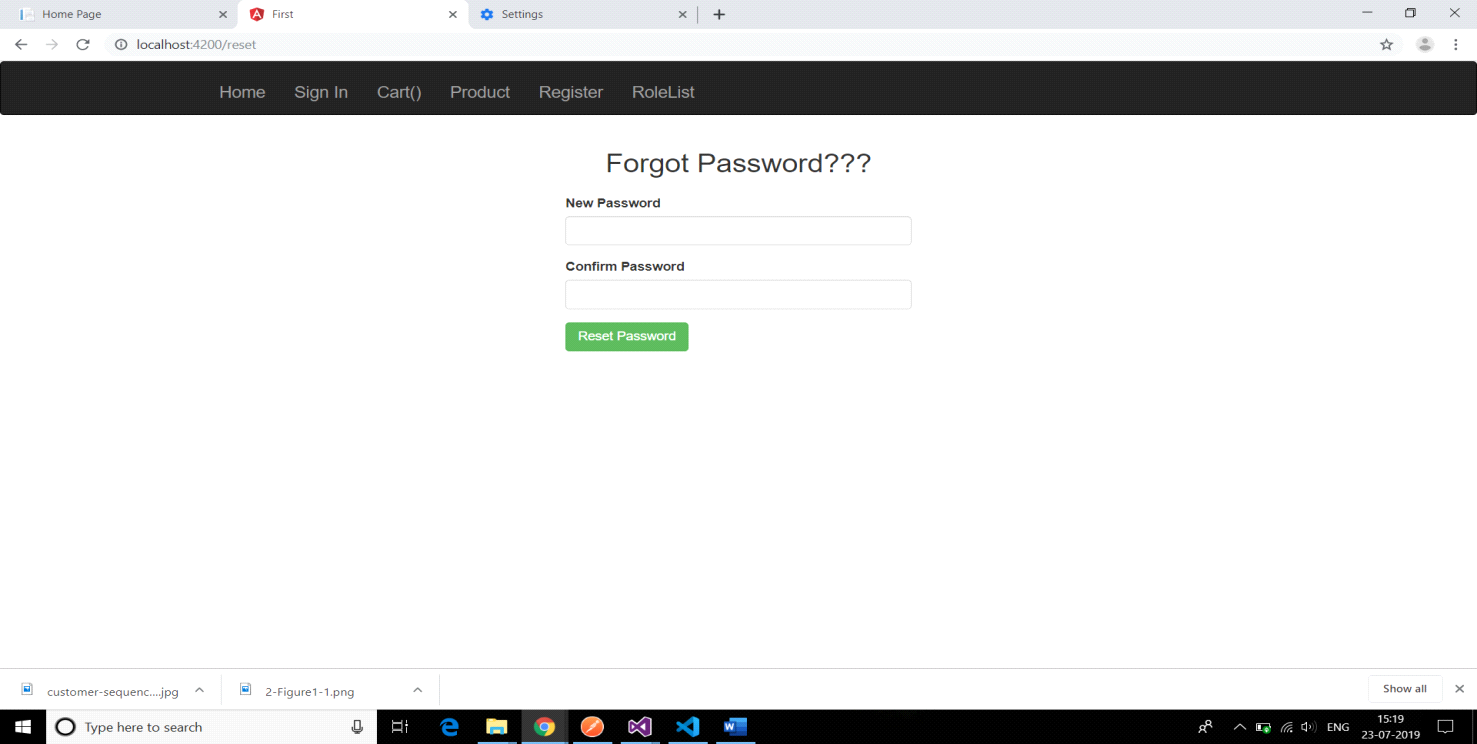
**Interface3:Registration**



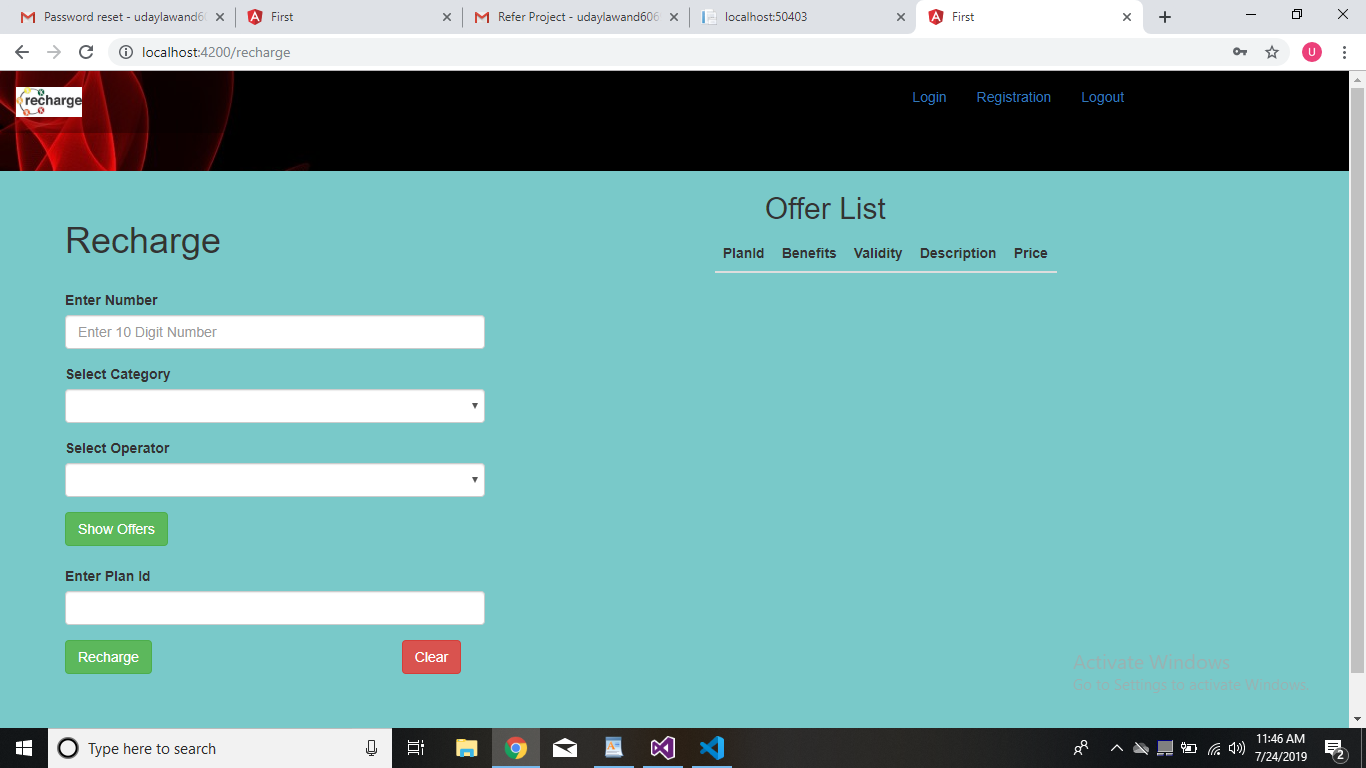
**Interface4:Forgot Password**



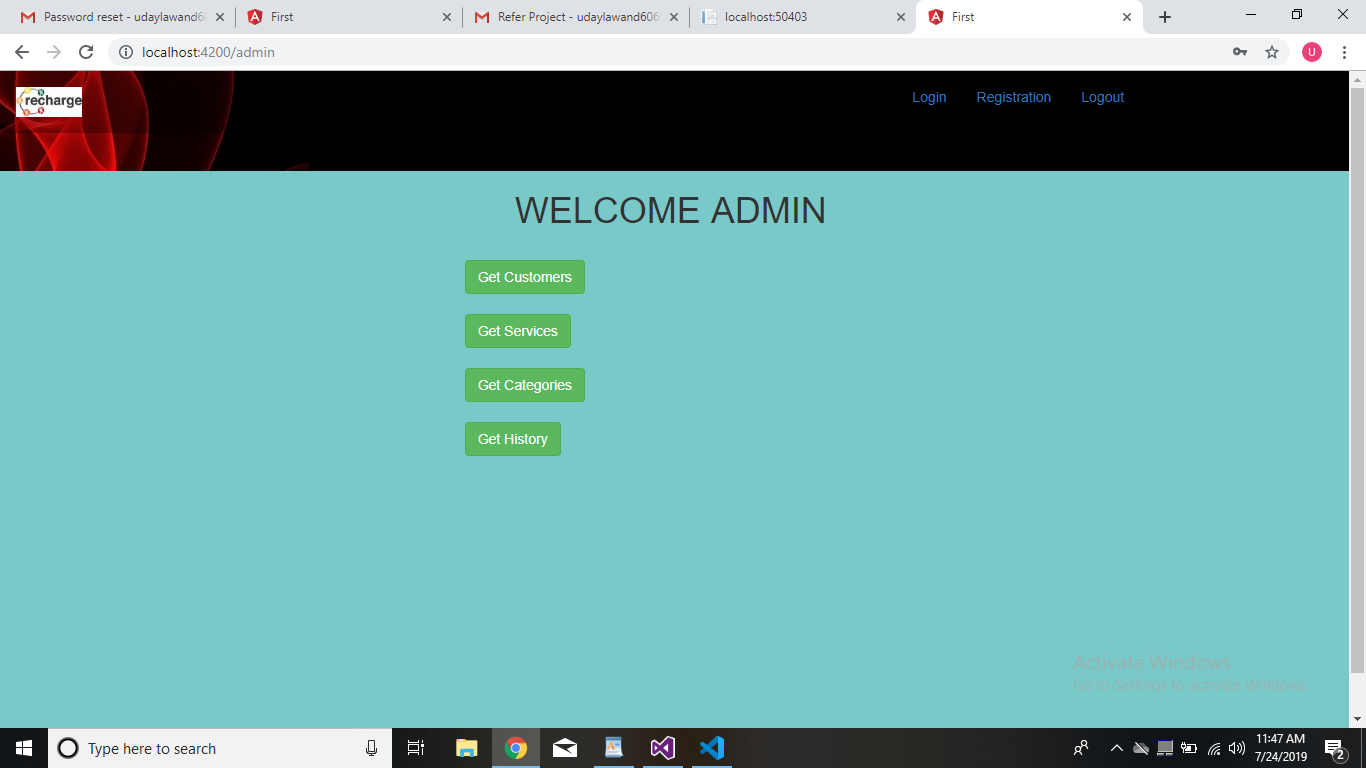
**Interface5:New password**



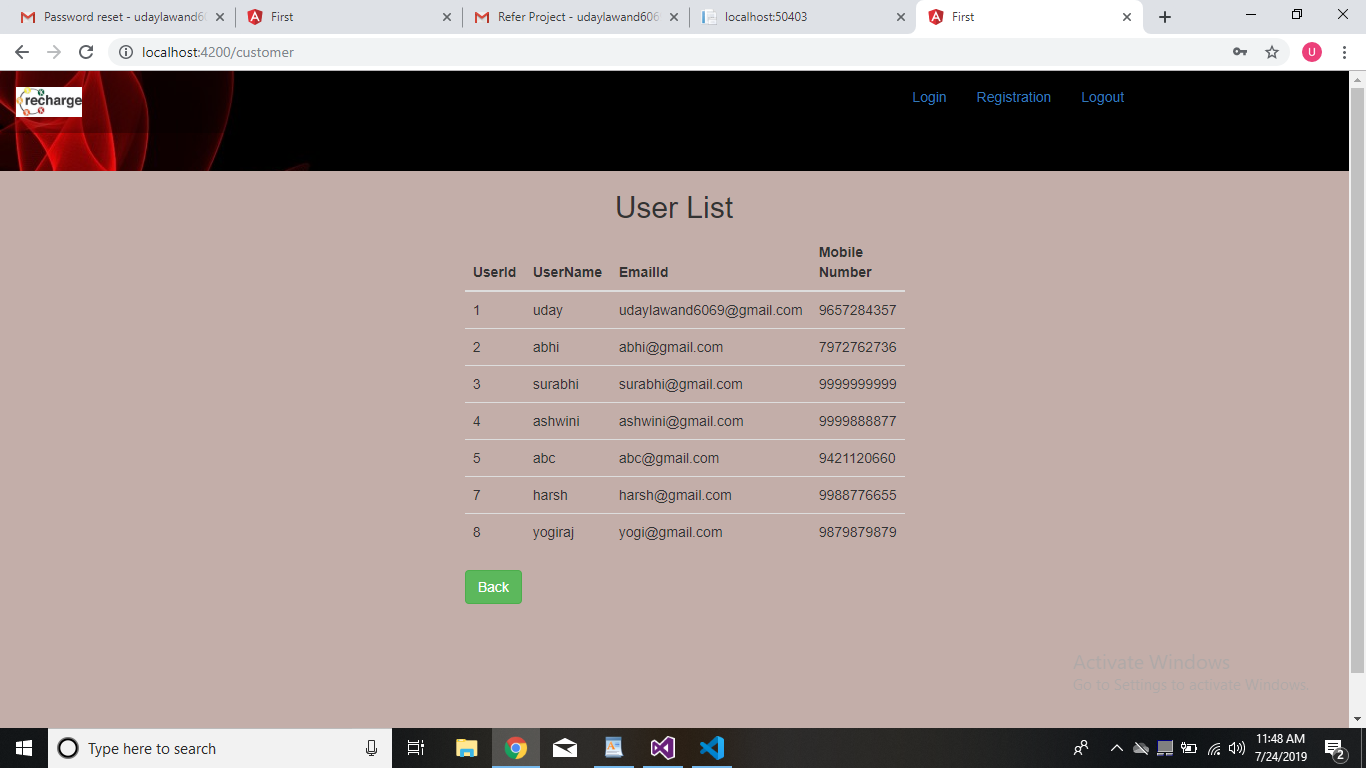
**Interface6:After user login**



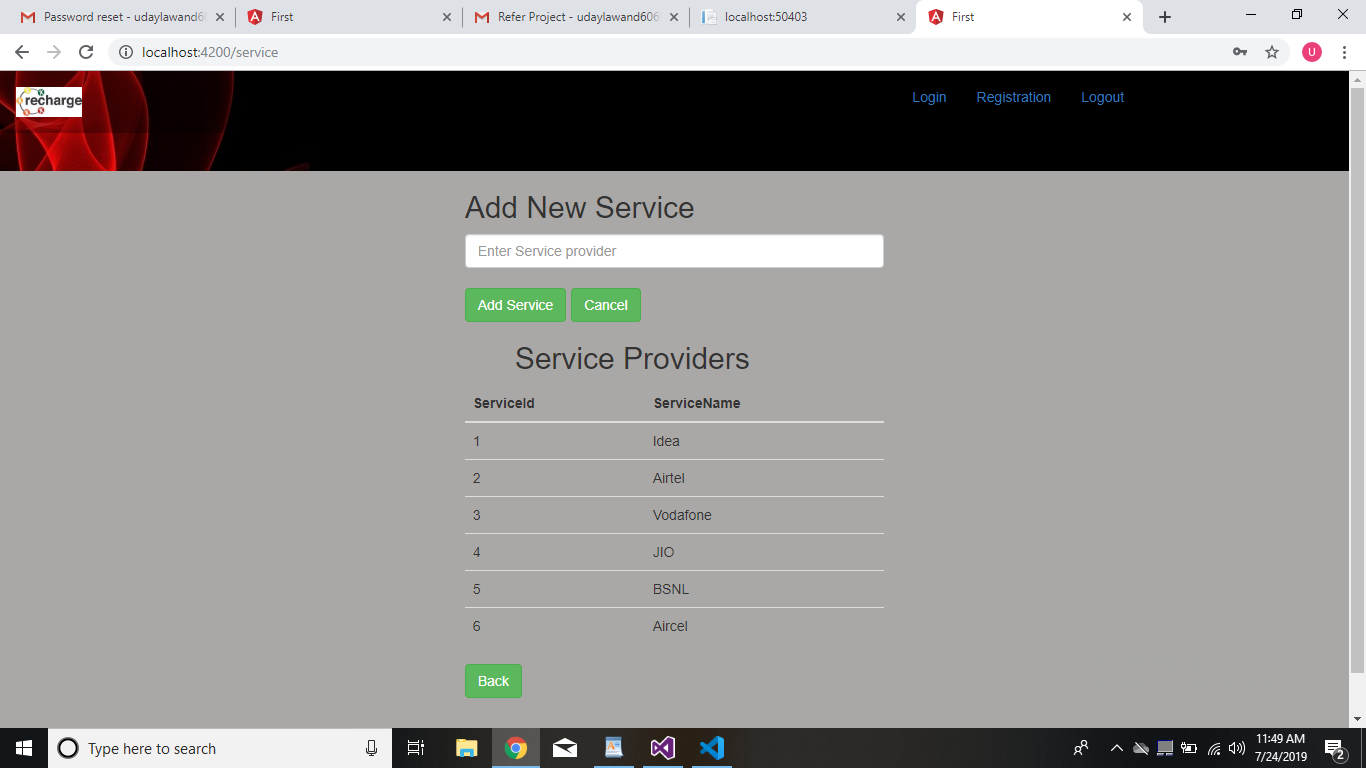
**Interface7:Admin page after admin login**



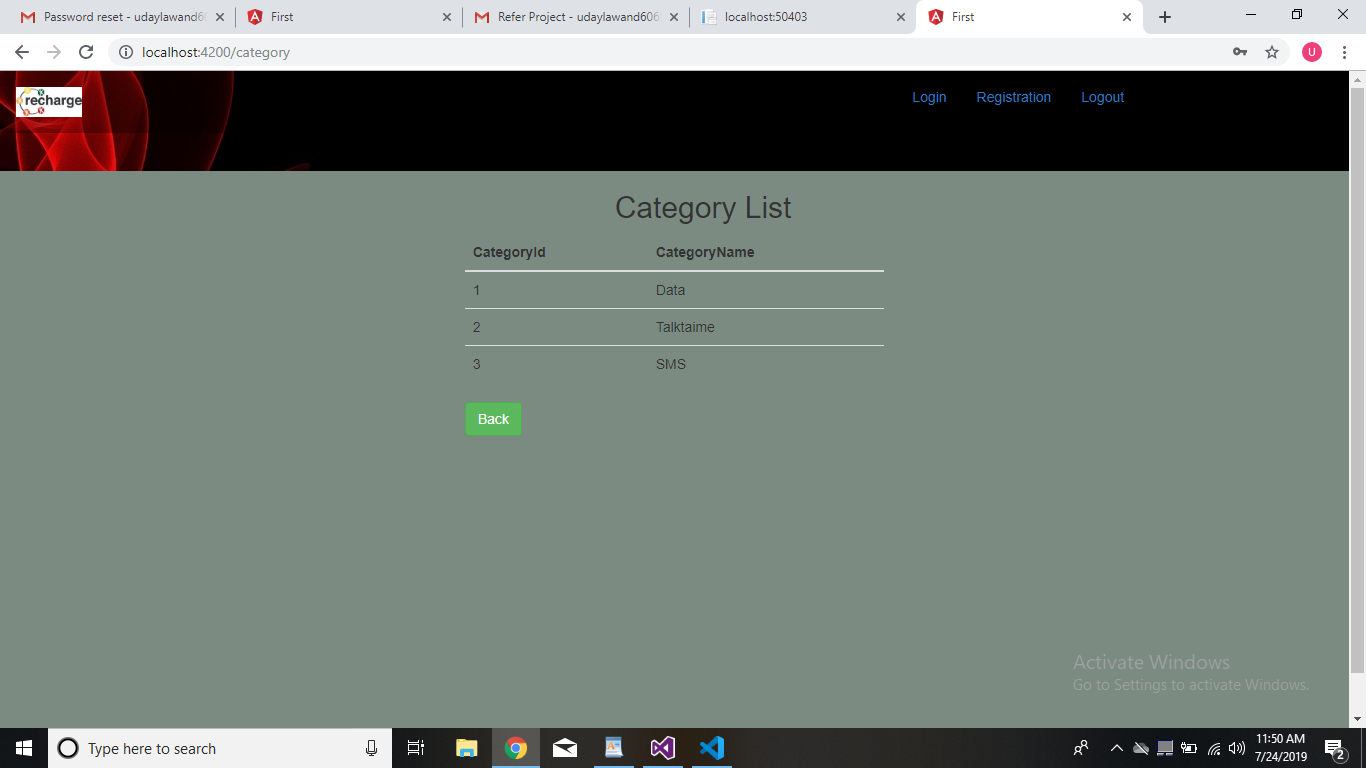
**Interface 8 : User List**



**Interface 9 :Services**



**Interface10 :Catagory**



**Interface 11 : History**

